

VHDL Design of eP32 Microprocessor

Silicon Valley Forth Interest Group

C. H. Ting

August 28, 2010

FORTH Microprocessor

- **Minimal instruction set:**
 - **Designs scalable from 16 to 64 bits**
- **Dual stack architecture:**
 - **Return stack for nested return addresses**
 - **Data stack for nested parameter lists**
- **Compute before execution:**
 - **All instructions executes in 1 clock cycle**
- **Minimized subroutine call and returns:**
 - **Support modular and structured programs**
 - **Seamless integration of high level programming language**

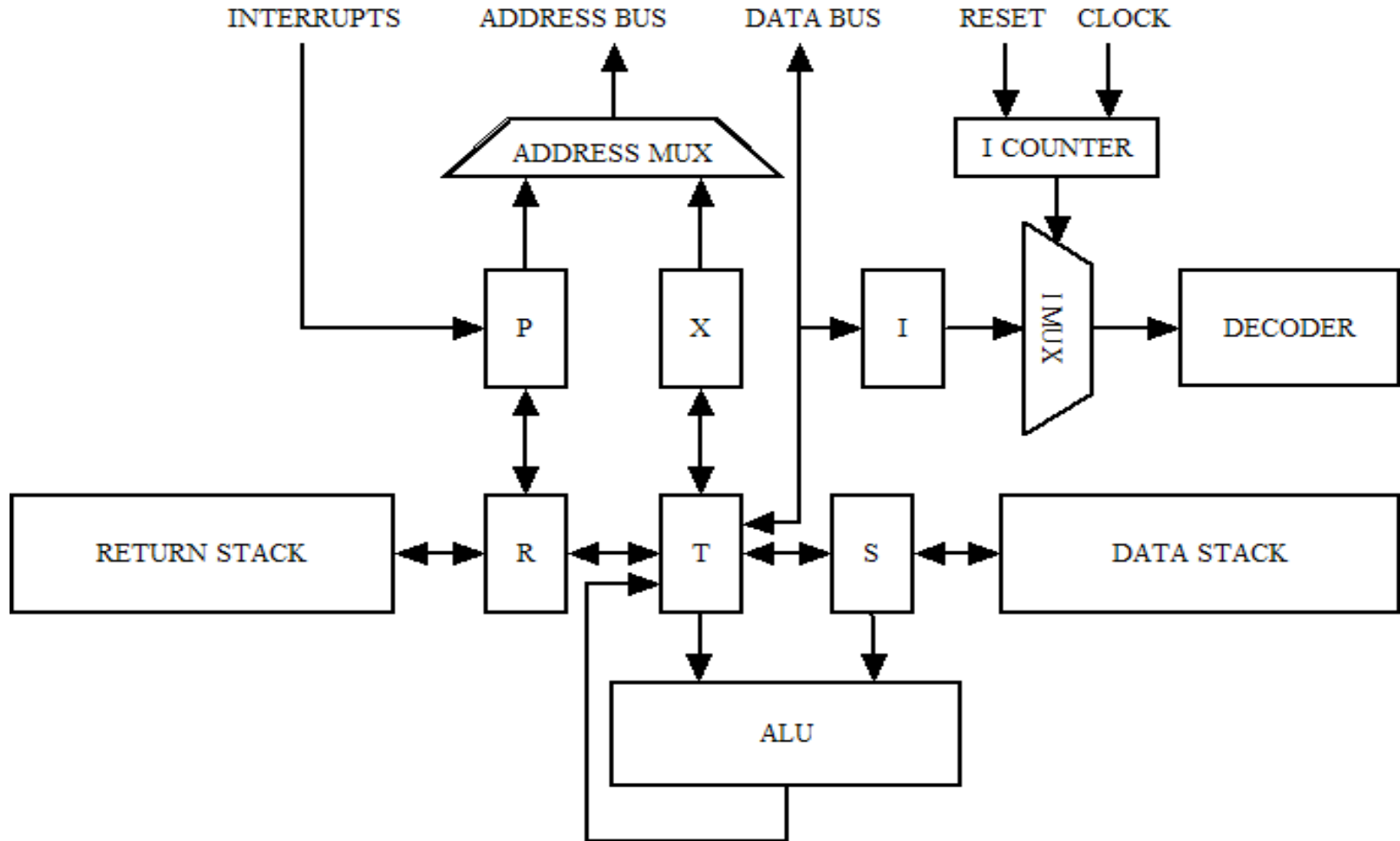
eP32 CPU Core

- **32 bit address and data busses**
- **25 powerful instructions extensible to 64 instructions**
- **32 level return stack**
- **33 level data stack**
- **Single cycle execution of all instructions**
- **Natural 5 instruction pipeline**

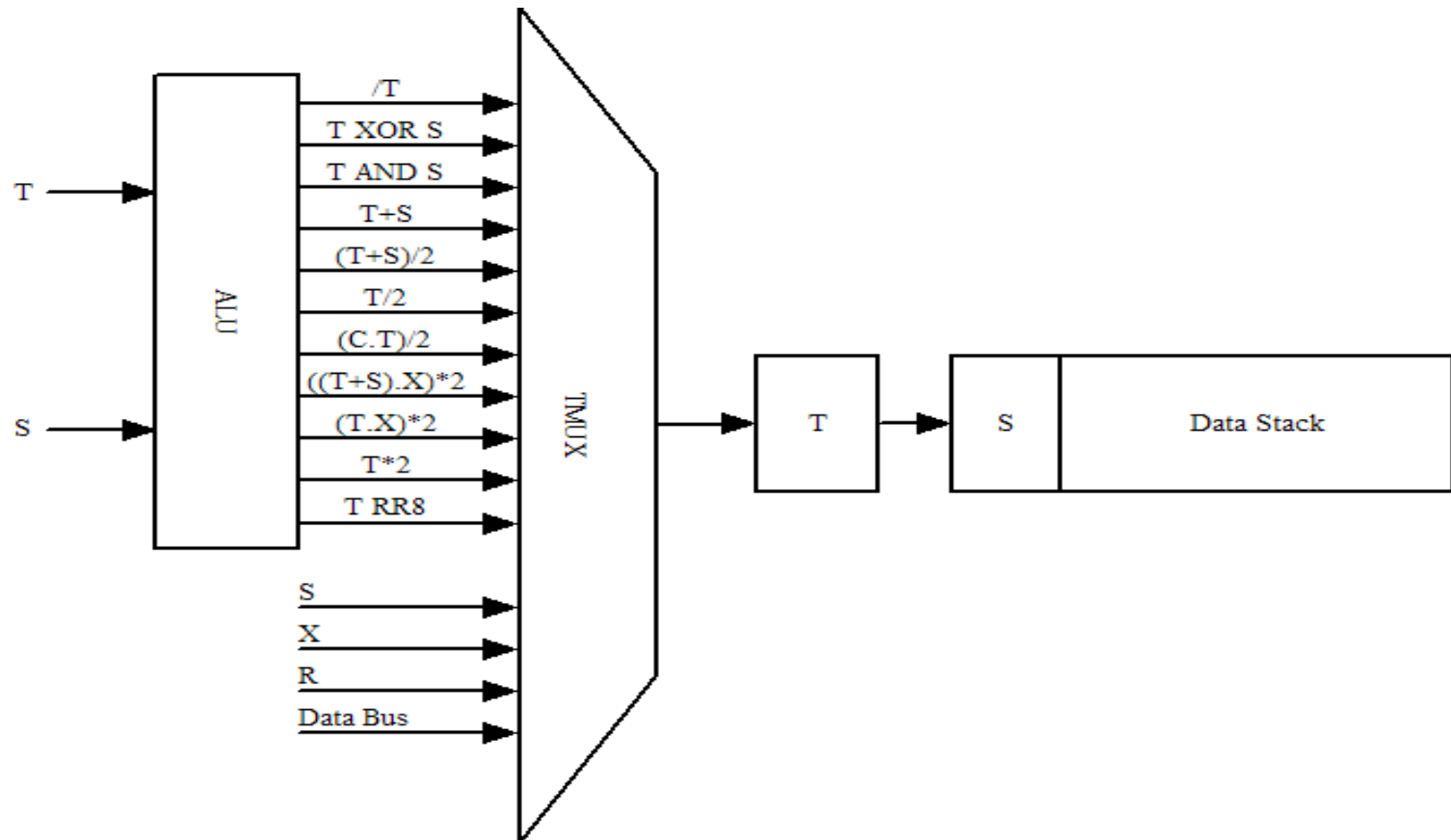
eP32 CPU Core

- **CPU architectural Overview**
- **ALU and data processing chain**
- **Program and data memory address multiplexer**
- **Return address processing chain**
- **Instruction execution finite state machine**

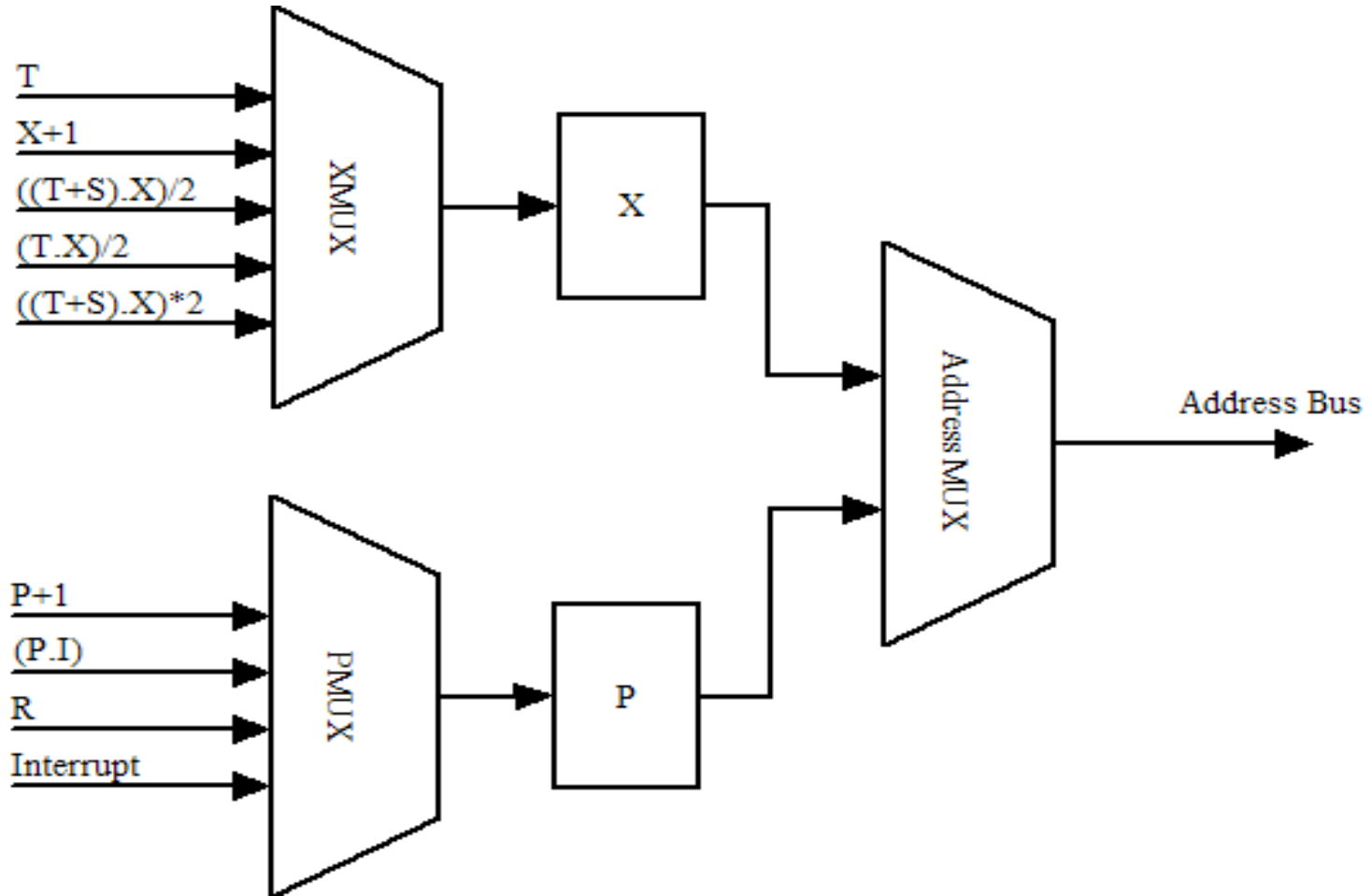
CPU Architectural Overview



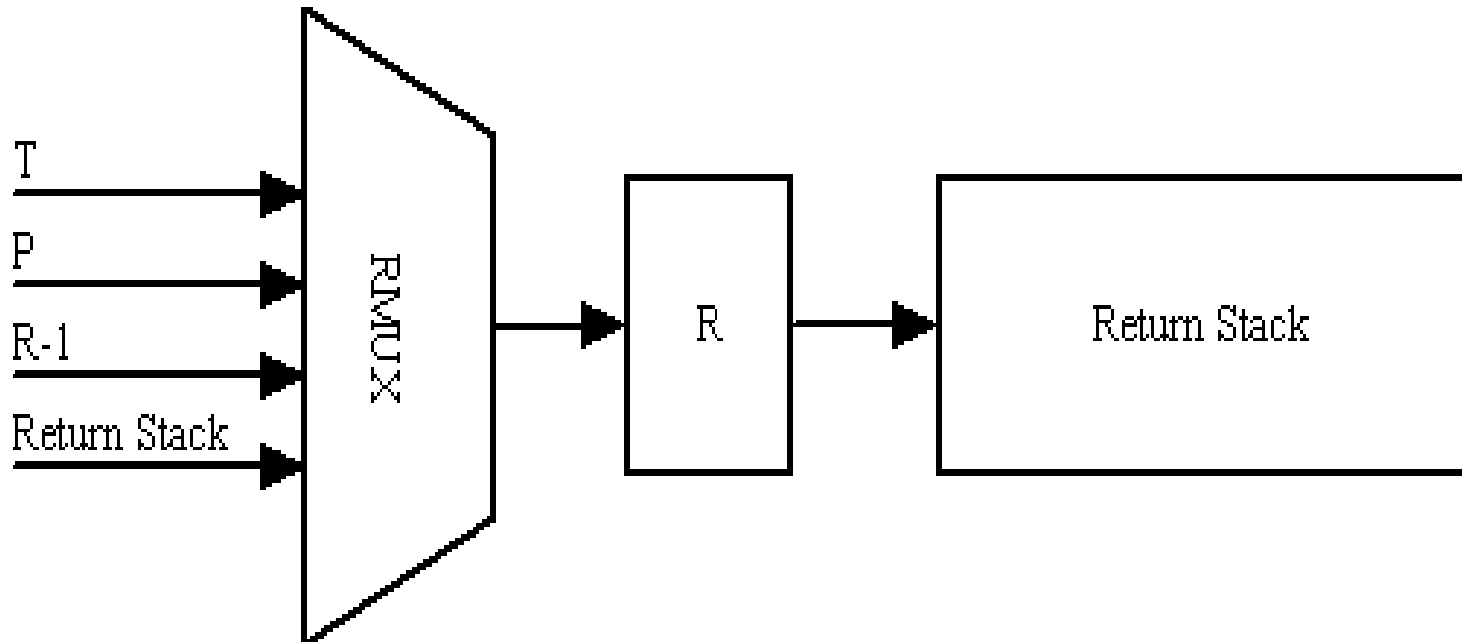
ALU and Data Processing Chain



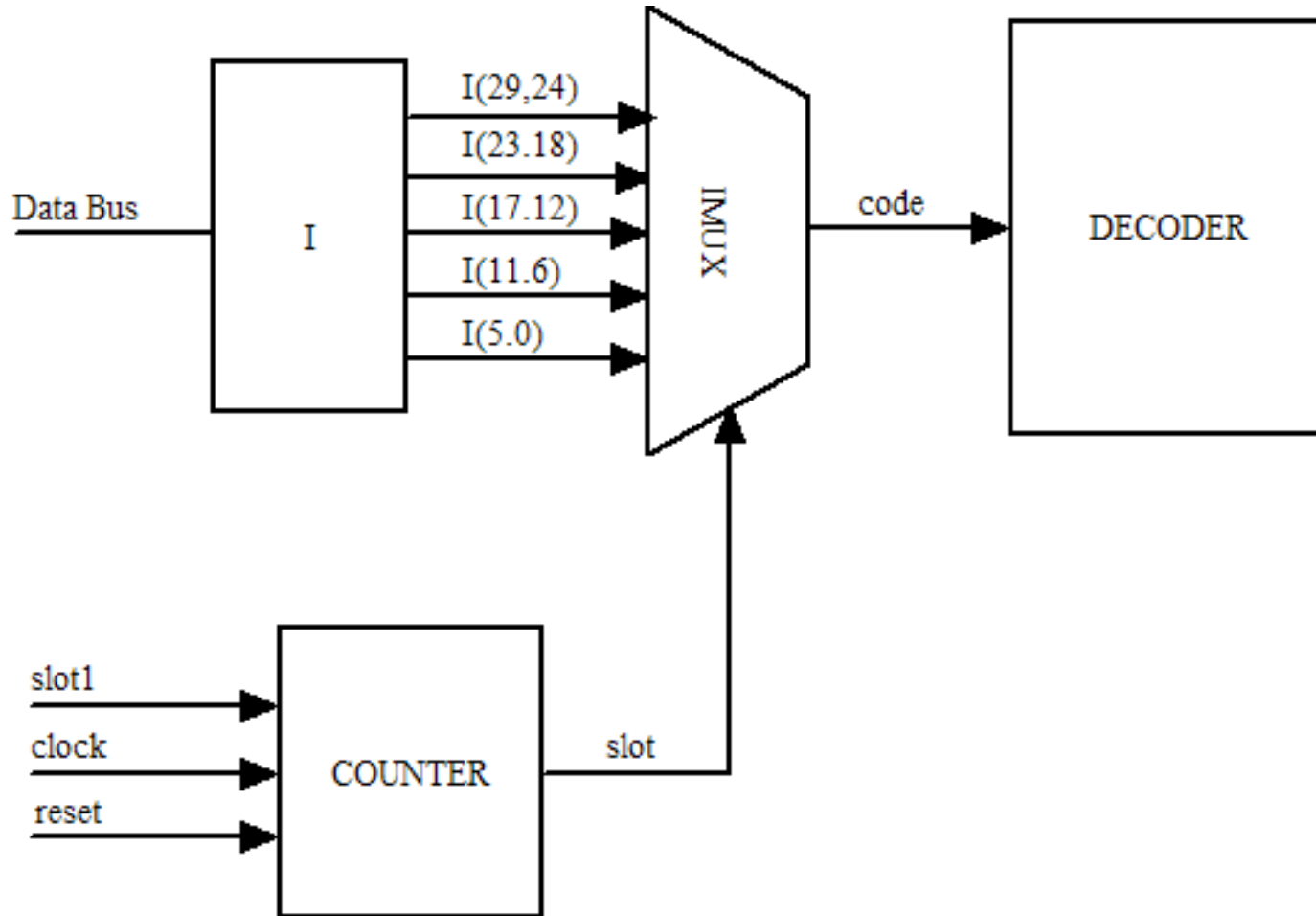
Program and Data Memory Mux



Return Address Processing Chain

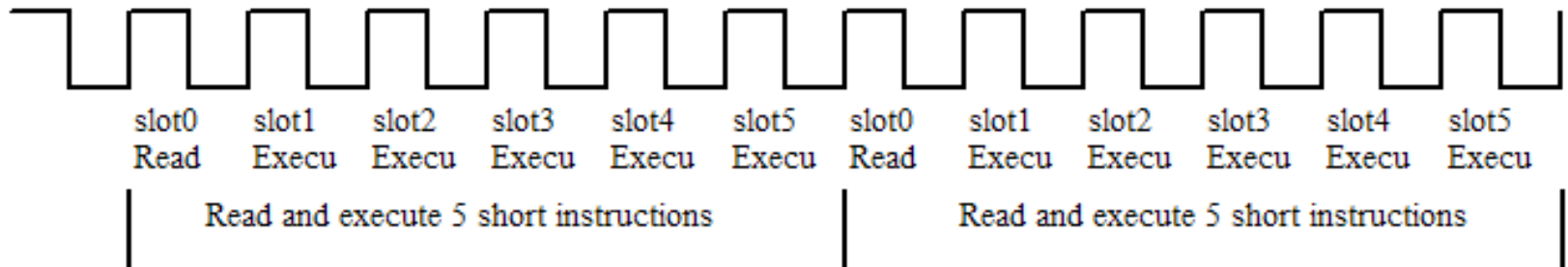


Instruction Execution FSM

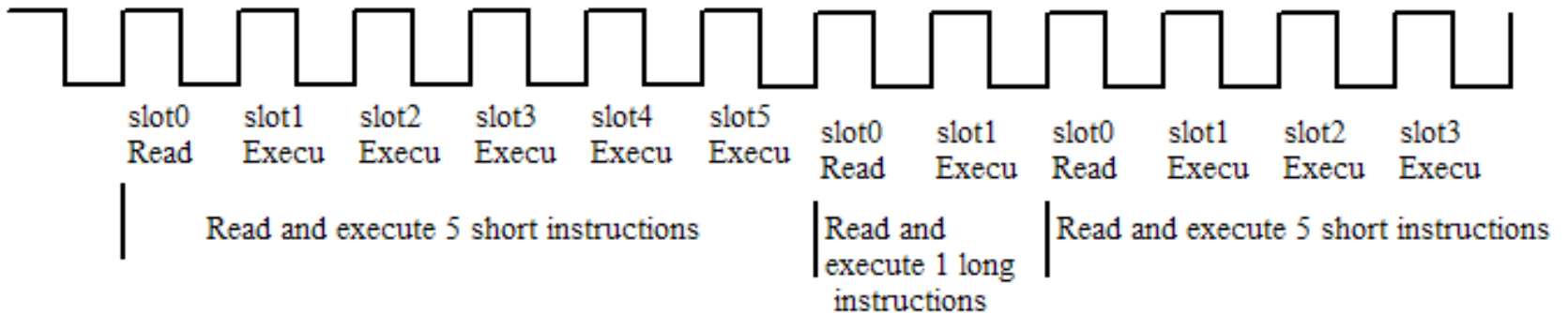


Instruction Execution Timing

Execution Cycles of Short Instructions



Execution Cycles of Long Instructions



eP32 Instruction Set

- **25 orthogonal instructions**
- **Encoded in 6 bit fields**
- **Easily expandable to 64 for specific applications**
- **4 Types of instructions:**
 - **6 Program transfer instructions**
 - **5 Memory access instruction**
 - **9 ALU instructions**
 - **8 Register and stack instructions**

Program Transfer Instructions

- **BRA** **Branch always**
- **RET** **Return from subroutine**
- **BZ** **Branch on zero**
- **BC** **Branch on carry**
- **CALL** **Call subroutine**
- **NEXT** **Loop until R is 0**

Memory Access Instructions

- **LD** **Load from memory**
- **LDP** **Load from memory and increment X register**
- **LDI** **Load immediate value**
- **ST** **Store to memory**
- **STP** **Store to memory and increment X register**

ALU Instructions

- **ADD** **Add S to T**
- **AND** **AND S to T**
- **XOR** **XOR S to T**
- **COM** **Complement T**
- **SHR** **T shift to right**
- **SHL** **T shift to left**
- **RR8** **T rotate right by 8 bits**
- **MUL** **Multiplication step**
- **DIV** **Division step**

Register and Stack Instructions

- **PUSHS** Duplicate T to S
- **POPS** Pop S to T
- **PUSHR** Push T to R
- **POPR** Pop R to T
- **OVER** Duplicate S over T
- **LDA** Load X to T
- **STA** Store T to X
- **NOP**

eP32 in VHDL

- **ep32q.vhd contains the complete source code in VHDL**
- **eP32 was implemented on these FPGA's:**
 - **Xilinx Virtex II**
 - **Actel ProASIC**
 - **Altera Stratix II**

Quartus Software System

- **Altera Stratix II FPGA chip for design and development**
- **eP32 system integrates:**
 - **eP32 CPU**
 - **RAM memory**
 - **UART**
 - **GPIO**
- **NIOS II Board for testing**

eP32 Forth System

- **CPU core**
- **32 levels of data and return stacks**
- **4K words of RAM**
- **UART**
- **16 bit GPIO**
- **50/16 MHz clock**

Synthesis Statistics

- **3368 Logic elements**
- **2473 Register**
- **131,072 Memory bits**
- **Synthesis time 5:55 minutes**

eForth Operating System

- **Subroutine thread model**
- **Word addressing**
- **Command interpreter**
- **High level command compiler**
- **Debugging utilities**

eForth Metacompiler

- **Based on F# eForth system**
- **Assembler**
- **Kernel**
- **Interpreter/Compiler**
- **Programming tools**
- **Simulator**

Demonstrations

- **Power-up NIOS II board**
- **Interactive eForth system**
- **Control LED indicators**
- **Operate switches**
- **Download and compile source code**

Thank you very much!

